

A SPARSE-TO-DENSE METHOD FOR 3D OPTICAL FLOW ESTIMATION IN 3D LIGHT-MICROSCOPY IMAGE SEQUENCES

Sandeep Manandhar¹, Patrick Bouthemy¹, Erik Welf², Philippe Roudot², and Charles Kervrann¹

¹Inria, Centre de Rennes - Bretagne Atlantique, France

²Department of Cell Biology, University of Texas Southwestern Medical Center, Dallas, USA

ABSTRACT

We present a two-stage 3D optical flow estimation method for light microscopy image volumes. The method takes a pair of light microscopy image volumes as input, segments the 2D slices of the source volume in superpixels and sparsely estimates the 3D displacement vectors in the volume pair. A weighted interpolation is then introduced to get a dense 3D flow field. Edges and motion boundaries are considered during the interpolation. Our experimental results show good gain in execution speed, and accuracy evaluated in computer generated 3D data. Promising results on real 3D image sequences are reported.

Index Terms— 3D PatchMatch, Interpolation, Fluorescence Microscopy, 3D Optical Flow

1. INTRODUCTION AND RELATED WORK

Estimating motion of cells and of subcellular particles is crucial for deciphering cell mechanisms and understanding cell behaviours. Modern 3D light microscopy (LM) for cell biology enables to observe cell dynamics at a good resolution both in space and time, motivating the development of 3D optical flow methods. However, the acquired 3D image sequences exhibit several specificities making 3D motion computation a difficult problem. The 3D stacks are not isotropic, the axial resolution is coarser than the lateral ones, and the stack usually comprises a small number of slices compared to the 2D image size. As a consequence, particles undergoing 3D movement might disappear in between two slices. Images are noisy; in most cases, due to the phototoxicity issue, the photon budget per image is reduced to favour an as long as possible acquisition over time. Finally, images are poorly textured, especially fluorescence image sequences which usually contain bright spots over cluttered background.

2D optical flow has been an active research field in computer vision for several decades. Optical flow methods estimate velocity components by relying on the brightness constancy constraint and imposing spatial smoothness of the velocity field, either locally or globally, while attempting to preserve motion discontinuities and to detect occlusions [1]. Recent performing methods proceed in two steps, computing a sparse field with an efficient matching method, especially to cope with large displacements and small moving objects, then interpolating the latter to recover a dense flow field as in [2]. Application of 2D optical flow methods to 2D fluorescence microscopy image sequences were reported in [3, 4].

3D optic flow computation has been far less investigated. A related problem is 3D image registration in medical imaging as surveyed in [5]. Let us also quote the local approach for 3D optical flow based on the all-pass filtering method defined in [6] and applied to 3D MRI data to handle respiratory motion. However, 3D registration in medical imaging mainly involves smooth deformation estimation over a dense isotropic volume, which is not the case for biological microscopy image stacks. Only a few investigations have been undertaken in living cell microscopy. Variational optic flow methods were tested for the first time in [7]. In [8], an algorithm is proposed where smoothness constraint is imposed by Markov Random Fields over the graph of presegmented super-voxels in the 3D volume. The method assumes that objects do not undergo complex motion, and computes a velocity vector per super-voxel. The method is used for the purpose of visualization by [9]. The authors of [10] integrate biophysical models in the estimation framework, and simultaneously compute force, pressure and velocity fields. The method, called BioFlow, originally applied to 2D microscopy image sequences, is extended to 3D data, and supplies convincing results on cell migration image sequences. Nevertheless, the method is effective if biophysical models are available, and takes about an hour for execution. In [11], local optical flow computed in the log-domain is used within a non-rigid registration framework based on diffeomorphic transformations and applied to 2D and 3D live cell image sequences. Computation time is reported to be around 33 minutes per frame.

In the 2D optical-flow paradigm, patch-matching based methods have been gaining state-of-the-art performance [2, 12, 13]. Following this idea and motivated by the design of a general-purpose fast 3D optical flow estimation method able to deal with

diverse 3D LM modalities and the variety of cell image contents, we propose a new 3D optical flow method for 3D LM image sequences. It is structured in two stages: sparse estimation and dense interpolation. For the first step, we develop a 3D version of PatchMatch [14] on supervoxels. The second step is achieved with a weighted interpolation scheme.

The remainder of the paper is organized as follows. In Section 2, we explain how we handle the 3D stack for matching. Section 3 describes our 3D PatchMatch method to compute the sparse velocity field. Section 4 presents the weighted interpolation to get the dense flow. In Section 5, we report and comment experimental results on 3D synthetic and real images. Concluding remarks are given in Section 6.

2. HANDLING OF THE 3D STACK

A LM image volume consists of a stack of 2D slices where each slice depicts information corresponding to a given depth layer of the cell. Considering this volume structure, we compute the 3D displacement vectors only at the points of the stack grid, that is, at the pixels of every slice grid. In what follows, the term displacement vector is interchangeably used as flow vector.

Our framework starts with the computation of a set of correspondences in a pair of LM image volumes I_1 and I_2 . To be efficient, matching must be performed on a sparse set of points. Superpixels are good candidates for that purpose, since we can easily control their density, they are almost regularly distributed over the image while taking into account the intensity patterns. We adopt the SLIC method [15]. We obtain K non overlapping superpixels in the slices of the volume I_1 . Here, supervoxels restrict to superpixels taken as their intersection with slice of the stack. Thus, we will still use the term superpixel in the sequel. Let S be the set of superpixels and s_k be the k^{th} superpixel.

For each s_k , we define a set of 3D neighbouring superpixels $\mathcal{N}_k = \{s_{k_i}, i = 1..n\}$ in I_1 which can lie in different slices. This neighborhood will be used in the propagation step of the 3D PatchMatch method as explained in the next section. \mathcal{N}_k comprises the n superpixels with minimum distance to s_k according to the similarity measure defined by:

$$d_{k,j} = \exp\left(-\frac{(c_j - c_k)^2}{\nu_d} - \frac{(I_1(c_j) - I_1(c_k))^2}{\nu_I}\right) \quad (1)$$

where c_k and c_i are the centres of superpixels s_k and s_j , and ν_d and ν_I are respectively the spatial distance and intensity value normalizers.

3. SPARSE MATCHING

Correspondences between points of I_1 and I_2 are defined as labels $\{l\}$ such that $I_1(x)$ and $I_2(x + l(x))$ form a matched pair, where label $l(x)$ is the 3D displacement vector of a voxel located at x in I_1 . To establish the correspondences, we opt for PatchMatch [14] which is a fast method to compute correspondence between two images. We extend Patchmatch to 3D volumes, where correspondences act between voxels.

The complexity of PatchMatch largely depends on the image grid size which is even bigger in case of volumetric data. We reduce the grid size by sparsely sampling the volumetric data into superpixels as explained in the preceding section. A similar strategy was used by [16] to compute approximate nearest neighbouring field (ANNF) between two 2D images. A smooth flow field suggests that similar voxels in a close vicinity move similarly. Then, we estimate a label for only the center c_k of superpixel s_k , as representative of its vicinity. A more sophisticated method is proposed in [17], introducing a specific patch neighborhood.

As with the original PatchMatch [14], the framework requires iterations of propagation and random search. For initialization, a label for c_k is chosen as the 3D vector to the center $c_{k'}$ of superpixel $s_{k'}$ in I_2 selected as the most similar one according to the distance (1), where $I_2(c_{k'})$ is substituted for $I_1(c_j)$. It proved to be better than the random initialization done in [14]. Then, we iterate propagation and random search. Scanning of the volume is performed in scan-line order in each slice, visiting each c_k , and slices are visited in a top-to-bottom way through the stack, as shown in Fig. 1.

We now briefly explain propagation. To update the label at c_k , we take the current labels of the centers of the superpixels in \mathcal{N}_k . This provides n possible correspondences for c_k . We select the label, among the n labels plus the current label of c_k , with a patch-based similarity measure. The patch is a local 3D patch of size $9 \times 9 \times 3$ around the point. For patch similarity, we adopt the L_2 norm of intensities between two patches. Other norms (L_1 , Lorentzian) did not improve results. After the best label \tilde{l}_k has been inferred for c_k in the propagation stage, the random search proceeds by first creating a search region of radius R centered at $c_k + \tilde{l}_k$ in I_2 . Then, a set of additional label candidates $\{l_r\}$ are sampled with an exponentially decreasing R through iterations. The superpixel-based sparse sampling allows us to take floating point label during the random search, which would have been computationally prohibitive otherwise. A new label among the l_r 's is assigned to c_k if the similarity is improved, compared to label \tilde{l}_k .

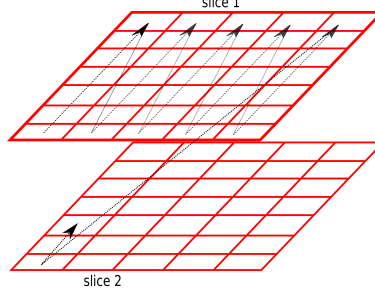


Fig. 1. Scanning order in the stack for 3D PatchMatch. The slices are regularly tessellated with cell size adapted to superpixel mean size. Each cell encapsulates a superpixel center.

At the very end of the alternate iterations, a sparse 3D flow field is obtained. A median filter is applied to the 3D vectors of the field to remove outliers that significantly disobey smoothness in the flow field. The filtered vectors are then used to interpolate displacements for the remaining voxels in I_1 as described in the next section.

4. SPARSE-TO-DENSE INTERPOLATION

To get a dense 3D flow field \mathbf{v} between I_1 and I_2 , the 3D displacement vectors for all the points in each superpixel segment need to be computed. To do so, we locally interpolate the sparse set of displacement vectors supplied by the first stage. Of course, we do not want to just duplicate the flow vector of c_k to all points of superpixel segment s_k , the undergone motion being not necessarily locally translational.

To this end, we design an interpolation scheme inspired by EpicFlow interpolation introduced in [2]. EpicFlow interpolation is an edge-aware interpolation technique that has been known to perform well in 2D optical flow benchmarks. It assumes that motion discontinuities are subset of intensity edges, and derives weights using an approximation of geodesic distance between points in Voronoi cells with intensity edges as cost map. In our case, we exploit the superpixel segmentation instead of Voronoi tessellation. In addition, motion boundaries will be taken into account in the interpolation stage in a different manner.

Let us consider a point x in superpixel s_k . The question is how to involve known vectors in the neighbourhood to interpolate the displacement or flow vector at x . For this, we introduce a *flow-aware* weight that encodes the respective orientation of known vectors. We allow interpolation with vectors of neighbours only when the flow vector at c_k and those of the superpixel centers of its neighbours in \mathcal{N}_k^* are converging. Here, \mathcal{N}_k^* is the subset of neighbouring superpixels lying in the same slice as s_k . Given the anisotropy of the volume, we do not involve off-the-plane superpixels.

We build an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where \mathcal{V} is the set of superpixel centers c_k 's, and \mathcal{E} is the set of edges between adjacent superpixels in \mathcal{S} . Let a and b be any two vertices (centers of superpixel s_a and s_b respectively). We define edge weights w_e using an approximation of geodesic distance between two vertices in \mathcal{V} defined by:

$$D_g(a, b) = \sum_{p_i \in L_{a,b}} q_{p_i}, \quad (2)$$

where p_i is a point in the line segment $L_{a,b}$ between a and b . q_{p_i} is the intensity edge response at p_i which can be derived from an intensity edge detector or simply from local intensity contrast. Noisy responses are filtered out.

We aim to nullify the interaction between two flow vectors if they are moving away from each other. Assuming no strong local curl in \mathbf{v} , we encourage interpolation from $\mathbf{v}(a)$ and $\mathbf{v}(b)$ to get flow vector at x in superpixel s_a , if the flow vectors $\mathbf{v}(a)$ and $\mathbf{v}(b)$ are converging as illustrated in Fig.2. Now, we define edge weights on \mathcal{E} as:

$$w_e(a, b) = \begin{cases} \frac{1}{D_g(a, b)}, & \text{if } \frac{d}{f} > 1 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where d is the Euclidean distance between a and b , and f the one between points $a + \mathbf{v}(a)$ and $b + \mathbf{v}(b)$ as shown in Fig.2b.

Given the weights derived from the respective orientation of the flow vectors and the spatial location of the superpixels, we interpolate for the flow vector at x as follows:

$$\mathbf{v}(x) = \frac{w_{a,x} \mathbf{v}(a) + \sum_{i \in \mathcal{N}_a^*} w_e(a, i) \mathbf{v}(i)}{w_{a,x} + \sum_{i \in \mathcal{N}_a^*} w_e(a, i)}, \forall x \in s_a, \quad (4)$$

where $w_{a,x} = 1/D_g(a, x)$. The interpolation of the 3D flow is performed slice by slice.



Fig. 2. Interpolation of flow vector at x (in red) according to the central flow vector (in purple) and the neighbouring flow vectors (in green). The purple and green points are the centers of their respective superpixels delineated by black lines.

Table 1. Error evaluation on 3D synthetic tests.

Method	Ours		Amat et al.[8]	
Error type	AEE	AAE	AEE	AAE
Translation	0.19	0.26	0.17	0.02
Rotation+Translation	1.74	0.14	1.93	0.11
Rotation+Scale	4.30	0.67	7.83	1.17

5. EXPERIMENTAL RESULTS

First, we objectively evaluated the method performance on computer-generated sequences with ground-truth, obtained by applying known transformations on real images issued from cell tracking challenge [18]. The image volume is of size $275 \times 231 \times 35$. 100 uniformly random transformations were applied for each class of transformation, from 3 to 10 units for x - and y -translation, from 2 to 3 units for z -translation, rotation range around the z -axis was $[-10^\circ, 10^\circ]$, and scaling range $[0.5, 3]$ in x and y , and $[0.5, 1.5]$ in z . We compared the 3D flow field obtained by our method against [8]. Table 1 contains average end-point error (AEE) and average angular error (AAE). Our method handles motion including rotation or scaling better than [8]. The latter method, unlike ours, computes a vector for a supervoxel, and assigns the same vector for the involved voxels which is not appropriate to handle complex motion. Figure 3b contains a portion of flow fields in a slice in the case of rotation and translation. The flow field computed by our method closely follows the ground-truth flow field.

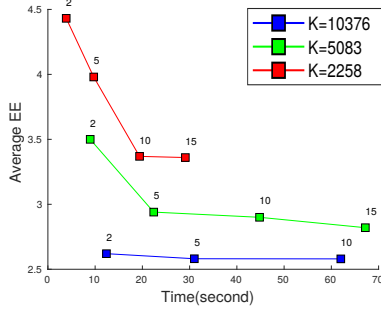
We have studied the tradeoff between runtime and accuracy with a varying number of superpixels. The method is implemented in C++ (no optimized) and run on a computer with 2.8 GHz Intel i7 processor and 16 GB of RAM. The parameters are set as follows: the size of initial random search region is $[40 \times 40 \times 7]$, $\nu_I = 0.3$ and $\nu_d = 100$. Plots in Fig.3a show a gain in accuracy with smaller size of superpixels (larger the K , smaller the superpixel size). The convergence is reached after 5-10 iterations (best viewed in electronic form for zooming or in color printed form).

We also applied our method in two real sequences [19, 20]. The first sequence depicts blebbing in a MV3 cell (Fig.4a-b). Blebbing is a phenomenon where cell membrane protrudes increasing the surface area of the cell. Protrusions, referred to as *blebs*, appear and disappear in interval of minutes. The appearance of a bleb is accompanied by stretching of a local region of the cell membrane. The region gradually retracts to adhere to conformity with the cell surface. Figure 5 plots the displacement vectors in one of the protrusions in the cell. The vectors tend to stretch the cell membrane. The computation time to estimate the 3D flow field between two volumes was 163s, with resp. 19s, 120s and 24s for superpixel generation, 3D patch matching, and interpolation.

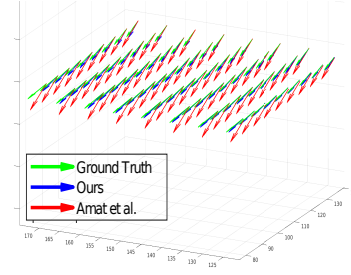
The second real sequence is a MV3 cell undergoing large scale protrusion depicted in Fig.4c-d. The main body of the cell in middle section squeezes as protrusions abruptly appear. A cross-section view of its 3D flow field is plotted in Figure 6 which suggests shrinking in the middle region and stretching in protrusions. The total computation time for one 3D flow field was 101 seconds, with 49, 44 and 8 seconds for resp. superpixel generation, 3D patch matching, and interpolation.

6. CONCLUSION

We have presented a method for fast 3D optical flow estimation in 3D LM image volumes. It involves the interpolation of a sparse 3D flow field obtained by matching superpixels per slice. We have developed a 3D PatchMatch algorithm to find correspondences. For densification, we have designed a weighted interpolation taking intensity edges and local flow variations into consideration. Tests performed on synthetic cases show its effectiveness, and experiments with real cases look promising. The 3D computed flow field could be further refined by post-processing it with a variational algorithm.

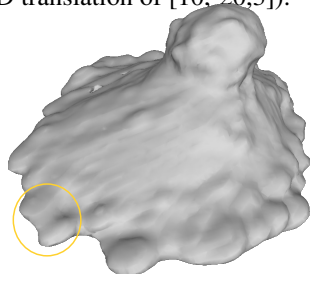


(a) AEE vs time in 3D.

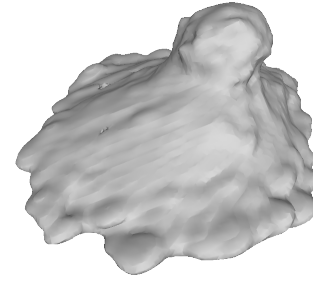


(b) 3D flow field in a slice.

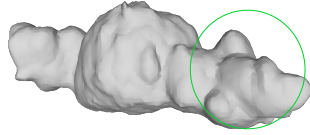
Fig. 3. Left: Accuracy converges to a value after few iterations (number of iterations is mentioned above each plotted points). Right: 3D flow vectors estimated in a slice with ground-truth in green, ours in blue, [8] in red (ground-truth: rotation of 10° around z -axis, 3D translation of $[10, -20, 3]$).



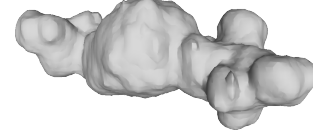
(a)



(b)



(c)



(d)

Fig. 4. (a) source and (b) target volumes for the first sequence with volume of size $241 \times 240 \times 101$, (c) and (d) for the second sequence with volume of size $256 \times 256 \times 143$ for two different MV3 cells. The colored circles roughly correspond to the regions highlighted in Fig.5 and 6 with respective color. Input microscopy images are courtesy of Danuser Lab, UTSW.

7. REFERENCES

- [1] D. Fortun, P. Bouthemy, and C. Kervrann, “Optic flow modeling and computation: a survey,” *Computer Vision and Image Understanding*, vol. 134, pp. 1–21, 2015.
- [2] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, “EpicFlow: Edge-preserving interpolation of correspondences for optical flow,” in *CVPR*, 2015.
- [3] J. Delpiano, J. Jara, J. Scheer, O. A. Ramirez, J. Ruiz del Solar, and S. Härtel, “Performance of optical flow techniques for motion analysis of fluorescent point signals in confocal microscopy,” *Machine Vision and Applications*, vol. 23, pp. 675–689, 2012.
- [4] D. Fortun, P. Bouthemy, P. Paul-Gillreux, and C. Kervrann, “Aggregation of patch-based estimations for illumination-invariant optical flow in live cell imaging,” in *ISBI*, 2013.
- [5] A. Sotiras, C. Davatzikos, and N. Paragios, “Deformable medical image registration: A survey,” *IEEE Trans. Medical Imaging*, vol. 32, no. 7, pp. 1153–1190, July 2013.
- [6] C. Gilliam, T. Küstner, and T. Blu, “3D motion flow estimation using local all-pass filters,” in *ISBI*, 04 2016.
- [7] Jan Hubeny, Vladimir Ulman, and Pavel Matula, “Estimating large local motion in live-cell imaging using variational optical flow,” in *VISAPP*, 2007.
- [8] F. Amat, E. W. Myers, and P. J. Keller, “Fast and robust optical flow for time-lapse microscopy using super-voxels,” *Bioinformatics*, vol. 29, no. 3, pp. 373–380, 2013.

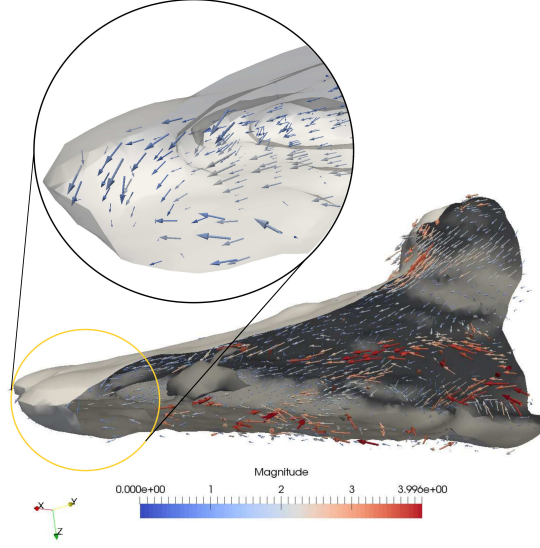


Fig. 5. Cross section view of flow field for the 1st sequence.

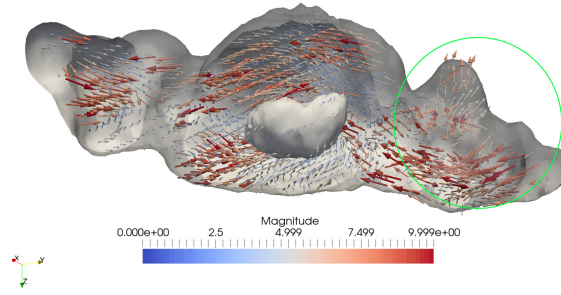


Fig. 6. Cross section view of flow field for the 2nd sequence.

- [9] C. P. Kappe, L. Schutz, S. Gunther, L. Hufnagel, S. Lemke, and H. Leitte, "Reconstruction and visualization of coordinated 3D cell migration based on optical flow," *IEEE Trans. Visualization and Computer Graphics*, vol. 22, no. 1, pp. 995–1004, 2016.
- [10] A. Boquet-Pujadas, T. Lecomte, M. Manich, R. Thibeaux, E. Labruyère, N. Guillén, J.-C. Olivo-Marin, and A. Dufour, "BioFlow: a non-invasive, image-based method to measure speed, pressure and forces inside living cells," *Scientific Reports*, vol. 7, no. 1, pp. 9178, Aug. 2017.
- [11] M. Tektonidis and K. Rohr, "Diffeomorphic multi-frame non-rigid registration of cell nuclei in 2D and 3D live cell images," *IEEE Trans. Image Processing*, vol. 26, pp. 1405–1417, 2017.
- [12] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine PatchMatch for large displacement optical flow," in *CVPR*, 2016.
- [13] C. Bailer, K. Varanasi, and D. Stricker, "CNN-based patch matching for optical flow with thresholded hinge loss," in *CVPR*, 2017.
- [14] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. on Graphics*, vol. 28, no. 3, pp. 24:1–24:11, July 2009.
- [15] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
- [16] J. Lu, Y. Li, H. Yang, D. Min, W. Y. Eng, and M. N. Do, "PatchMatch filter: Edge-aware filtering meets randomized search for visual correspondence," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1866–1879, 2017.
- [17] R. Giraud, V.T. Ta, A. Bugeau, P. Coupé, and N. Papadakis, "SuperPatchMatch: An algorithm for robust correspondences using superpixel patches," *IEEE Trans. Image Processing*, vol. 26, no. 8, pp. 4068–4078, Aug. 2017.
- [18] M. Masaka, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. España, S. Venkatesan, D. M. Balak, P. Karas, T. Bolcková, M. X0160 Treitova, C. Carthel, S. Coraluppi, N. Harder, K. Rohr, K. E. Magnusson, J. Jaldén, H. M. Blau, O. Dzyubachyk,

P. Krízek, G. M. Hagen, D. Pastor-Escuredo, D. Jimenez-Carretero, M. J. Ledesma-Carbayo, A. Muñoz-Barrutia, E. Meijering, M. Kozubek, and C. Ortiz-de Solorzano, “A benchmark for comparison of cell tracking algorithms,” *Bioinformatics*, vol. 30, no. 11, pp. 1609–1617, 2014.

- [19] K.M. Dean, P. Roudot, E.S. Welf, G. Danuser, and R. Fiolka, “Deconvolution-free subcellular imaging with axially swept light sheet microscopy,” *Biophysical Journal*, vol. 108, pp. 2807–2815, 2015.
- [20] K.M. Dean, P. Roudot, C.R. Reis, E.S. Welf, M. Mettlen, and R. Fiolka, “Diagonally scanned light-sheet microscopy for fast volumetric imaging of adherent cells,” *Biophysical Journal*, vol. 110 (6), pp. 1456–1465, 2016.